#include <stdio.h>

#include <gd.h>

#include <error.h>

#include <omp.h>

int main(int argc, char \*argv[])

{

int nt = 4;

int tid,tmp,red,green,blue,color,x,h,y,w;

tmp=red=green=blue=color=x=h=y=w=0;

char \*iname =NULL;

char \*oname = NULL;

gdImagePtr img;

FILE \*fp={0};

if(argc!=3)

error(1,0,"format : object\_file input.png output.png");

else

{

iname = argv[1];

oname = argv[2];

}

if((fp=fopen(iname,"r"))==NULL)

error(1,0,"error : fopen : %s",iname);

else

{

img = gdImageCreateFromPng(fp);

w=gdImageSX(img);

h=gdImageSY(img);

}

double t=omp\_get\_wtime();

omp\_set\_num\_threads(nt);

#pragma omp parallel for private(y,color,red,blue,green,tmp) schedule(static)/\*schedule(dynamic,50) schedule(guided,50)\*/

for(x=0;x<w;x++)

for(y=0;y<h;y++)

{

tid= omp\_get\_thread\_num();

color=gdImageGetPixel(img,x,y);

red=gdImageRed(img,color);

green=gdImageGreen(img,color);

blue=gdImageBlue(img,color);

tmp=(red+green+blue)/3;

red=green=blue=tmp;

color=gdImageColorAllocate(img,tmp,tmp,tmp);

gdImageSetPixel(img,x,y,color);

/\*if(tid==0)

{

color=gdImageColorAllocate(img,red,0,0);

gdImageSetPixel(img,x,y,color);

}

if(tid==1)

{

color=gdImageColorAllocate(img,0,green,0);

gdImageSetPixel(img,x,y,color);

}

if(tid==2)

{

color=gdImageColorAllocate(img,0,0,blue);

gdImageSetPixel(img,x,y,color);

}

if(tid==3)

{

color=gdImageColorAllocate(img,0,0,0);

gdImageSetPixel(img,x,y,color);

}\*/

}

t=omp\_get\_wtime()-t;

printf("\ntime taken : %lf threads : %d\n",t,nt);

if((fp=fopen(oname,"w"))==NULL)

error(1,0,"error : fopen : %s",oname);

else

{

gdImagePng(img,fp);

fclose(fp);

}

gdImageDestroy(img);

return 0;

}

//run gcc -fopenmp image.c -lgd

//./a.out in.png out.png